Submitted to the Journal of Web Semantics
draft material -- please do not cite

The Foundational Model of Anatomy in OWL: experience and perspectives

Christine Golbreich[1], Songmao Zhang[2], Olivier Bodenreider[3]

[1]LIM, University Rennes 1, 35043 Rennes, France
Christine.Golbreich@univ-rennes1.fr

[2]Institute of Mathematics, Chinese Academy of Sciences, Beijing 100080, P.R. China
smzhang@math.ac.cn

[3]U.S. National Library of Medicine
8600 Rockville Pike, MS 43, Bethesda, Maryland 20894, USA
olivier@nlm.nih.gov

Abstract. We present the method used for migrating the Foundational Model of Anatomy (FMA) from its representation with frames in Protégé to its logical representation in OWL and our experience in reasoning with it. In spite of an extensive use of metaclasses in Protégé, converting the FMA from Protégé into OWL DL was possible and most of its original features were captured. The conversion relies on a set of translation and enrichment rules implemented with flexible options. Unsurprisingly, reasoning with the FMA in OWL proved to be a real challenge, due to its sheer size and complexity, and raised inference problems hard to solve in terms of time and memory. However, various smaller versions have been successfully handled by Racer. Some inconsistencies were identified and several classes reclassified. The results obtained so far show the advantage of OWL DL over frames and, more generally, the usefulness of DLs reasoning techniques for building and maintaining the large-scale biomedical ontologies of the future Semantic Web.

1 Introduction

Life Sciences have a long tradition of controlled vocabularies. Huge terminologies, classifications and ontologies have been developed for many years in various biomedical domains. These resources have the potential to contribute to the Semantic Web for the Life Sciences, but need to be adapted for it. A large library of biomedical ontologies has been developed in frames, often with Protégé [12]. As OWL is the W3C recommended standard for ontologies [1], converting frame-based ontologies to OWL becomes an important need. Representing ontologies in OWL provides several advantages. Once converted to OWL, ontologies currently developed with frames become virtually *interoperable* with other ontologies and can be used as resources for the Semantic Web. Interoperability is important for shared use across different biological and medical domains, as expected for example from the Open Biomedical

Ontologies (OBO) library. Also of interest is OWL higher expressiveness, and precise formal semantics. Another important advantage of OWL is the existence of *powerful reasoning services*, based on its underlying description logic. Several major ontological and terminological resources in biomedicine, e.g., the Medical Subject Headings (MeSH) [8], the Gene Ontology™ [9] and the National Cancer Institute Thesaurus [10] are being converted to OWL DL. The conversion of other ontologies to OWL have also been investigated, including the UMLS® Metathesaurus® [2] and Semantic Network [11]. A long term goal is to provide a Web service assisting the conversion of frame-based ontologies to OWL, in order to take advantage of the higher expressiveness and powerful reasoning services of its underlying description logic and to facilitate interoperability. At the moment, the present goal is to investigate the conversion of a large frame-based ontology into OWL and the reasoning services enabled by this conversion.

The frame-based ontology under study is the Digital Anatomist Foundational Model of Anatomy (FMA). It was converted from Protégé 2.1 to OWL DL. The FMA is the most complete ontology of human 'canonical' anatomy [3]. The version used in this conversion, dated of July 2004, contains *70,169 concepts and more than 1.5 million relations*. The FMA was selected for several reasons. First, anatomy plays a prominent role in biomedicine and many biomedical ontologies and applications refer to anatomical ontologies. As its authors claim, the FMA is "a *reference* ontology in biomedical informatics for correlating different views of anatomy, aligning existing and emerging ontologies in bioinformatics ..." [3]. Anatomy, together with Gene and Disease reference ontologies, constitute the backbone of the future Semantic Web for Life Sciences. Thus, *interoperability* is a main motivation in migrating the FMA from frames to OWL. Second, the FMA in OWL is a real challenge from a knowledge representation perspective. Indeed, it was interesting to investigate if OWL DL, which is a first order language, had a sufficient expressiveness to represent what was originally represented with frames and metaclasses in Protégé. On the other hand, evaluating OWL and DL inference techniques for real large-scale biomedical ontology such as the FMA was really attractive. Indeed, the sheer size and complexity of the FMA, with its 70,169 concepts and 1.5 million relations, make it a real challenge for DLs reasoning techniques and OWL tools, both for editors (e.g., Protégé OWL) and reasoners (e.g., Racer). As a main goal of the FMA conversion to OWL was to evaluate the advantages of DLs over frames and the possible benefits obtained from reasoning with OWL, the language selected for the conversion is OWL DL, in contrast to OWL Full proposed in [5]. Indeed, OWL DL provides completeness and decidability of the interesting reasoning problems (satisfiability and subsumption) supporting consistency checking and automatic classification. OWL DL reasoners are available (e.g., Racer [16] and Pellet [17]), while OWL Full is undecidable, offers no computational guarantees and lacks suitable reasoners.

The rest of this paper is organized as follows. The method used to automatically convert the FMA from Protégé 2.1 into OWL DL is first presented (§2). Our experience in reasoning with OWL is reported next (§3). The choices of conversion, as well as possible perspectives for the FMA and open questions for large-scale ontologies of the future Semantic Web are finally discussed (§4).

2 Conversion to OWL DL

As DLs and frames share the same object paradigm, it might be thought that converting a Protégé frame-based[1] ontology into OWL is straightforward and could be achieved by a simple export function mapping Protégé primitives to OWL constructs. But, the export function from Protégé to OWL did not work for the FMA, neither in one step (i.e., directly), nor in two steps (i.e., from database to text then to OWL). Besides, even if it had worked, it would be ineffective, mainly for two reasons:

First, migrating a frame-based ontology to OWL requires not only a syntactic "translation", but also a semantic "enrichment". Indeed, *property restrictions* such as `allValuesFrom` or `someValuesFrom` contained in the OWL axioms cannot be directly derived from the original frame representation, where they are not specified. Additionally, classification strongly relies on the classes *logical* definitions. A reasoner (e.g., Racer) can only automatically classify classes under "defined" classes[2] – i.e., classes with at least one necessary and sufficient condition. Necessary *and* sufficient conditions cannot be derived directly from the Protégé model neither, because in frames, all slots at class with a specified range or value, are considered as a set of necessary conditions. Specifying *defined classes* is a major "enrichment" of the ontology.

The second reason is that the frame-based representation of the FMA in Protégé makes extensive use of metaclasses[3], which are not allowed in the first order language OWL DL. Each anatomical entity is modeled both as a metaclass and as an instance of a metaclass. This was the "*technical solution for enabling the selective inheritance of attributes*" in Protégé [3] (see discussion §4). For example (Table 3 of Annex), `Heart` is defined as a metaclass, subclass of `Organ_with_cavitated_organ _parts`, itself subclass of `Organ`, and as an instance of this metaclass. At the meta level, `Heart` inherits all the slots, facets, characteristics (range, cardinality, inverse etc.) of its superclasses. `Heart` inherits from `Organ` the slot `bounded_by` with multiple values allowed in `Surface_of_organ`, the slot `arterial_supply` with multiple values allowed in the classes `Artery`, `Arteriole_Arterial_plexus` or `Set_of_arteries`, the slot `venous_drainage` with multiple values in the class `Subdivision_of_venous_tree_organ` or `Organ_part_tree_structure`, etc. But at the class level, the own slots of `Heart` are assigned particular values, e.g., `bounded_by` is filled with `Surface_of_heart`, `arterial_supply` with `Right_coronary_artery` and `Left_coronary_artery`, etc. Directly translating metaclasses into OWL would lead to OWL Full, instead of OWL DL. Simply removing metaclasses as suggested in [5] would not be satisfactory either, since the knowledge encoded at the metaclasses would be lost.

---

[1] In the following, 'Protégé' is used as a shortcut for Protégé-Frames (not for Protégé-OWL). Protégé-Frames (version 2.1) and Protégé-2000 is a tool for building *frame-based* ontologies. Protégé-OWL (current versions 3.1.1 or 3.2) is a Protégé extension that supports OWL.

[2] except if a property has a domain (or range) that is a primitive class, which can coerce classes to be reclassified under the primitive class that is the domain or range of the property (§4).

[3] A metaclass is a class whose instances are themselves classes

Therefore, we defined our own method of conversion to OWL DL, which aims at providing the desired enrichments and at capturing the knowledge that was encoded at metaclasses, but differently.

2.1 Method of conversion

The migration was achieved from the CLIPS files. The conversion relies on *translation (i)* and *enrichment (ii)* rules. Different enrichment rules were defined depending on whether the information was defined at the *class* or *metaclass (iii)* level. The conversion rules are implemented with flexible options (the rules used in the current version can be found in the Annex).

(i) Translation draws on the structural correspondence between Protégé-Frames primitives and OWL constructs. The Protégé class taxonomy defined at meta level is translated into an OWL subclass hierarchy. Template slots defined at the top level are translated into OWL properties with the same features as those specified in Protégé (i.e. same range, inverse, cardinality, etc.), simply mapping each of them to the corresponding OWL primitive (Fig. 1). For example, the Protégé single slots 'has_mass' and 'has_boundary', defined with type SYMBOL, allowed values FALSE TRUE, and cardinality 0 1, are simply translated into an owl:DatatypeProperty, with range datatype Boolean, and declared to be an owl:FunctionalProperty. The Protégé multislot constitutional_part defined with type SYMBOL, allowed parents Physical_anatomical_entity and inverse slot constitutional_part_of, is translated into an owl:ObjectProperty with (rdfs:range rdf:resource="#Physical_anatomical_entity") and inverse (owl:inverseOf rdf:resource="#constitutional_part_of").

Protégé slot	OWL property
Type INTEGER, FLOAT, STRING	*DatatypeProperty* with range datatype integer, float and string
Type SYMBOL with allowed values T or F	*DatatypeProperty* with range datatype boolean
Type SYMBOL with allowed values (not T nor F)	*ObjectProperty* with range the enumerated class of all the allowed individuals
Type SYMBOL with allowed parents	*ObjectProperty* with range the union of all the allowed classes
Type INSTANCE with allowed classes	

Fig. 1 Some translation rules for slots

(ii) Enrichment, in contrast, introduces new logical features. The enrichment rules were designed to reflect the underlying principles of the original FMA model. Some enrichment rules and the rationale behind them are presented below.

– **Property restrictions**: the choice between universal and existential property restrictions is mainly based on the distinct role of template and own slots in Protégé. Template slots "specify which slot each member of a class shall have and what the restrictions (facets) on the values of these slots shall be" [3]. Template slots with their constraints are inherited by the subclasses and the instances. Therefore, allowed parents or allowed classes specified for a template slot at metaclass, are converted

into *universal* property restrictions (`owl:allValuesFrom`). In contrast, according to the FMA principle of "canonical anatomy" [3], when a class instantiates a metaclass, the specific values assigned to a template slot inherited as own slot describe the typical canonical structure of the particular anatomical entity in terms of relations that should necessary exist, e.g. in terms of the existing parts composing an organ. Therefore, they are converted into *existential* property restrictions (`owl:someValuesFrom`) (Fig. 2).

Protégé template slot at metaclass	**OWL property restriction**
Slot with allowed parents or allowed classes C_i	*owl:allValuesFrom* property restriction constrained to the union of all the classes C_i
Slot with an allowed value	*owl:hasValue* property restriction constrained to the specified value
Protégé own slot at class	*owl:someValuesFrom* property restriction constrained to the class C
Slot assigned with a specific class C as value	
Slot assigned with a specific datatype or individual value	*owl:hasValue* property restriction constrained to the specified value

Fig. 2 Some enrichment rules

For example, the multislot `bounded_by` of the metaclass `Organ` with allowed-parents `Surface_of_organ` is converted into the universal restriction (∀ `bounded_by Surface_of_organ`) on the property `bounded_by` of `Organ`, that is next inherited by its subclass `Heart`. But when `Heart` inherits `bounded_by` as an own slot assigned with the value `Surface_of_heart`, it is converted into the existential restriction (∃ `bounded_by Surface_of _heart`).

Similarly, `venous_drainage` is restricted by a universal restriction inherited from its superclasses, but when `Heart` inherits `venous_drainage` as an own slot assigned with the values `Oblique_vein_of_left_atrium`, `Left_marginal_vein`, `Coronary_sinus`, `Posterior_vein_of_left_ventricle`, `Unnamed_tributary_-of_cardiac_vein`, `Anterior_interventricular_vein`, `Small_cardiac_vein`

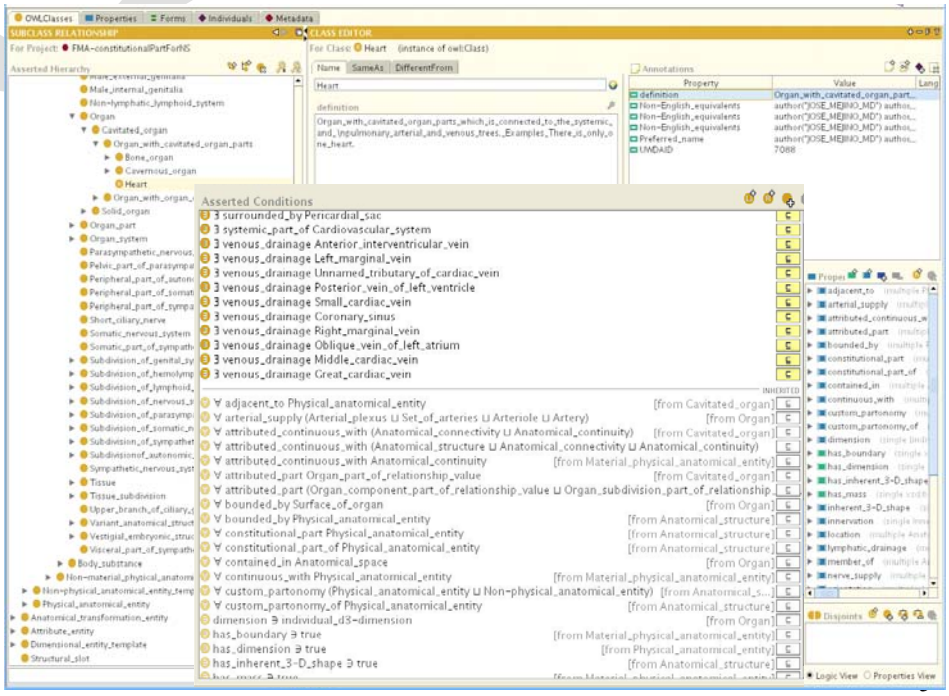


Fig. 3 Class Heart and some of its asserted or inherited properties restrictions of

etc. they are converted into `owl:someValuesFrom` restrictions specifying the value constraints on the property for the class `Heart` (Fig. 3).

– **Equivalent class definition**: a "defined" class has at least one necessary and sufficient condition. At this preliminary step, one slot *p* is manually selected, and a class *A* having values $B_1,\ldots, B_n$ assigned to its own slot *p,* is defined as equivalent to the conjunction of all the existential value restrictions on *p* to the classes B_i and of metaclass and superclass of *A* (after some optimization). As aggregated objects are often described in terms of their parts and as meronymic relationships play a particularly important role in anatomy, it was chosen to define anatomical entities in terms of their parts. At this first step, the property "constitutional part" was selected, resulting in 570 defined classes. Thus, the equivalent class expression used for the defined classes combines taxonomic relations, metaclass instantiation and constitutional parts that were defined in the original FMA model. For example, the class `Heart` is defined by the conjunction `Organ_with_cavitated_organ_parts` ⊓ (∃ `contitutional_part Wall_of_heart`) ⊓ (∃ `contitutional_part` … as shown Fig. 4.

The choice of the property 'constitutional part' was partly motivated by a size issue: constitutional part is relatively well populated in the FMA, compared for instance to 'custom partonomy', thus is computationally more significant. But, such a definition is not "semantically" satisfying for all classes: all the anatomical entities cannot be uniformly defined solely in terms of their constitutional parts (the same parts may belong to different structures), and no such constitutional parts are defined for most FMA classes. But at this step, the priority was to test if a DL reasoner, here Racer, could be run for consistency checking and classification. Different definitions for the different subtrees, and more complex expressions combining several properties shall be next investigated (§ 4).

Fig. 4 The defined class Heart

(iii) Metaclasses are converted into ordinary OWL DL classes.

– First, subclass relation between metaclasses and metaclass instantiation are both translated into OWL `subClassOf` axioms ([A] of B in Protégé is converted to A ⊏ B in OWL). Doing so, the metaclass and instance definitions are merged within a single class in OWL DL (Fig. 5).

– Second, the restrictions depend on whether the information was defined at the *class* or *metaclass* level, according to the enrichment rules (cf. ii). Range restrictions of a template slot defined at metaclass are converted to *universal* property restrictions.

– Third, the values of class own slots are converted either into existential restrictions on class properties or into values of OWL annotation properties (cf.

Annex). Structural own slots with values assigned at class are converted to *existential* restrictions of object or datatype property. Own slots such as definition, name, identifiers, e.g., UWDAID, etc. with values assigned at class, are converted to OWL `AnnotationProperty`, e.g., `definition, prefered_name, UWDAID,` (Fig. 3) instead of using metaclass instances, which prevents such properties from being propagated to their instances or subclasses.

Thus, each entity of the FMA is represented by a single OWL DL class, with various axioms and annotations. Fig. 5 shows the OWL DL class `Heart` with its `equivalentClass` definition combined with `subClassOf` axioms including universal or existential restrictions, derived from the original metaclass and class definitions of `Heart` in Protégé.

In conclusion, although the original frame-based representation in Protégé made extensive use of metaclasses, converting the FMA into OWL DL is possible. The OWL version of FMA complies with OWL DL's first order language constraints. In particular a class is not an individual at the same time. In fact, the 'higher' order structure in Protégé frames was deleted in replacing metaclass instantiation by subclass axiom. But it did not introduce significant change, as "all concepts in the FMA Anatomy Taxonomy are subclass of a superclass and also an instance of a metaclass". The representation of `Heart` in frames and in OWL DL is presented Fig. 5. As explained later (§4.1), nothing has been lost, and it was not necessary to give up some knowledge of the original FMA, because OWL DL higher expressiveness enabled to capture the intended meanings of the FMA metaclasses.

Heart in Protégé frames	**Heart in OWL DL**
Metaclass (defclass Heart (is-a Organ_with_cavitated_organ_parts) ...) Instance of Metaclass ([Heart] of Organ_with_cavitated_organ_parts (constitutional_part Wall_of_heart Cavity_of_left_atrium Cavity_of_right_ventricle Cavity_of_left_ventricle Right_coronary_artery Left_coronary_artery … (bounded_by Surface_of_heart) (arterial_supply Right_coronary_artery Left_coronary_artery) …)	<owl:Class rdf:ID="Heart"> <owl:equivalentClass> <owl:Class> <owl:intersectionOf rdf:parseType="Collection"> <owl:Class rdf:about= "#Organ_with_cavitated_organ_parts"/> <owl:Restriction> <owl:onProperty rdf:resource="#constitutional_part" /> <owl:someValuesFrom rdf:resource="#Wall_of_heart" /> </owl:Restriction> … </owl:equivalentClass> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="#bounded_by"/> <owl:someValuesFrom rdf:resource="#Surface_of_heart"/> </owl:Restriction> </rdfs:subClassOf> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="#arterial_supply"/> <owl:someValuesFrom rdf:resource="#Right_coronary_artery" /> </owl:Restriction> … </owl:Class>

Fig. 5 Class Heart in OWL DL, derived from the Protégé metaclass and class definitions

To compensate for the arbitrariness of some of our choices, the enrichment rules were designed and implemented with flexible options. This permitted to automatically

generate various OWL files with different flavors, size and computational complexity. Moreover, these options can be easily modified, which is key to the incremental approach adopted for reasoning (§3.1).

2.2 Results

In order to minimize the size of the resulting ontology, (only inessential) laterality distinctions were ignored, i.e. classes differing from their parents *only* by laterality. The resulting subset comprises about 40,000 concepts, i.e. 57% of the 70,000 concepts of the original FMA. Applied to this subset, the conversion process described earlier resulted in about 117,000 frames, including 40,000 OWL named classes. More precisely, there are 187 properties and 85 individuals specified in this file. 20 properties correspond to annotation, 19 to datatype and 148 to object properties. There are 107,238 `subClassOf` axioms (38,772 from taxonomy and 3,378 from metaclass instantiation), 39,337 classes where 559 are defined by `equivalentClass` axioms. OWL constructors `allValuesFrom`, `someValuesFrom`, `hasValue`, `oneOf`, `unionOf`, `FunctionalProperty`, `SymmetricProperty`, `InverseOf` all occur in the OWL file resulting from the conversion (available at http://mor.nlm.nih.gov/pubs/supp/2005-owled-cg/FMA-constitutionalPartForNS.owl) It took about 15 minutes to load the FMA OWL file in Protégé-OWL (version 2.1[4]) in a Windows XP PC with 4GB memory (90 minutes with 512Mb).

3 Reasoning with OWL

Reasoning with OWL proved to be a real challenge, due to the sheer size and complexity of the FMA. The full-fledged FMA in OWL DL raised inference problems hard to solve in terms of time and memory, so an incremental approach was adopted for reasoning.

3.1 Incremental approach

We used Racer (Version 1.7) with the OWL files generated by the conversion process to investigate consistency checking and automatic classification. Launched from Protégé-OWL, the classification failed for the entire FMA. Running Racer directly from Rice, we experienced problems related to memory limitation (4GB). Since Racer could not handle the entire FMA OWL file (in fact restricted to 2/3 of the whole FMA), as suggested by the Racer authors, we decided to test smaller versions so as to reduce the size and time issues and to investigate eventual errors, adding more features incrementally. First, a FMA OWL version with all classes but without any properties was checked to test if the taxonomy alone could be successfully classified. Then, we added equivalent class definitions using only one property to test if the

[4] 4 minutes in version 3.1

ontology with defined classes could pass Racer. Next, we successively introduced, step by step, object properties, annotation properties, datatype properties, and finally object properties used for attributed slots. When properties are introduced in partial versions, the conversion rules described previously are applied. For example, a small version where the object property `bounds` and its inverse `bounded_by` are introduced includes, for each class having these properties specified, the subclass axioms containing the corresponding existential and universal restrictions of the properties `bounds` and `bounded_by`.

3.2 Results

Racer passed the first test: the classification of the FMA OWL version without any properties was successful, taking about 25 minutes with 512Mo memory on a Pentium 4. Then, the classification with 'defined' classes described by the conjunction of the existential restrictions on the `constitutional_part` or `custom_partonomy` property as necessary and sufficient condition was also successful. Next, various versions were generated with all classes but containing a limited number of properties. Depending on the properties introduced and the constructors involved, the tests were successful or not. Some results of tests are summarized below[5]:

Reasoning with Racer was *successful* for the following partial versions:

1. Ontology with only the class hierarchy defined but without any property.
2. Ontology with defined classes (based on `constitutional_part`).
3. Ontology with defined classes and annotation properties.
4. Ontology with defined classes, annotation properties, and all datatype properties.
5. Ontology with defined classes and primitive classes with restrictions on the property `branch_of` in subClassOf axioms.
6. Ontology with defined classes and primitive classes with restrictions on the property `arterial_supply` in subClassOf axioms.
7. Ontology with defined classes and primitive classes with restrictions on the property `2D_part` in subclass axioms.
8. Ontology with defined classes and primitive classes with restrictions on the property `bounds` and its inverse `bounded_by` in subclass axioms.
9. Ontology with defined classes and primitive classes with restrictions on properties `dimension` and `has_physical_state` in subclass axioms.
10. Ontology with primitive classes with restrictions on attributed slot `location` and all slots used in `location` (e.g., `related_object`, etc.).

[5] the test files will be made available at http://mor.nlm.nih.gov/download/fma_owl/

11. Ontology with primitive classes with restrictions on attributed slot `attributed_part` and all slots used in `attributed_part` (e.g., `related_part`, etc.)

Reasoning with Racer *failed* for:

12. Ontology with defined classes and annotation properties, added with primitive classes with restrictions on all the object properties in subclass axioms.
13. Ontology with defined classes and primitive classes with restrictions on all object properties in subclass axioms.
14. Ontology with primitive (resp. defined) classes with restrictions on the property `branch_of` and on its inverse in subclass axioms.
15. Ontology with Subclass axioms with restrictions on the property `continuous_with`, declared symmetric.

The results of the tests are summarized in the tables below, showing the OWL DL constructs (Table 1) and axioms (Table 2) involved in each test (denoted by '•'). A is a class name, C is a class, o is an individual name, R is an object property, T is a datatype property, D is a data range, v is a data value, and l; m; n are integers (according to usual notations borrowed from OWL authors [1]). `AnnotationProperty` (used for annotations) are omitted in the tables, but tests #3, #4 and #12 all include them. As there was no hierarchy of relationships specified in the original frame-based FMA, equivalent or sub-property axioms including `equivalentProperty` and `subPropertyOf` were not defined in our conversion.

The reasons for failure are not easy to analyze. For instance, Racer was successful for some tests including inverse properties, such as test #8 having equivalent class axioms based on `constitutional_part` and restrictions on the property `bounds` and its inverse `bounded_by`, or a test with `constitutional_part` and its inverse `constitutional_part_of`, while it failed for some tests including inverse, such as test #14 having restrictions on the property `branch_of` and its inverse `branch` (however, which does not use nominal, cardinality or functional axioms). This experience shows that the sheer size of the FMA is not the only issue. The results of reasoning with OWL for the FMA are related to several factors, including the complexity of the generated ontology due to the OWL constructors used, i.e. the presence of nominals (e.g., `oneOf`), of `inverseOf` axioms or "global" axioms, etc. and their interactions. The origin of the computational difficulties encountered is not completely clear at the moment. On the one hand, some inference problems hard to solve might come from imperfect conversion. On the other hand, theoretical work has proved that reasoning is NExpTime-complete for OWL DL ($\mathcal{SHOIN}$), and precisely inverse, nominals, number restrictions, anonymous classes all occur in the FMA ontology in OWL. The algorithms and optimization techniques implemented by the reasoners, are certainly critical issues for the FMA.

	Test														
# Test	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Success (S) or Failure (F)	S	S	S	S	S	S	S	S	S	S	S	F	F	F	F
Classes Constructors															
intersectionOf($C_1 \ldots C_n$)		•	•	•	•	•	•	•	•			•	•		
unionOf($C_1 \ldots C_n$)		•	•	•	•	•	•	•	•	•	•	•	•	•	•
complementOf(C)															
oneOf($o_1 \ldots o_n$)									•	•	•	•	•		
Restriction R															
allValuesFrom(C)		•	•	•	•	•	•	•	•	•	•	•	•	•	•
someValuesFrom(C)		•	•	•	•	•	•	•	•	•	•	•	•	•	•
hasvalue(o)									•			•	•		
[minCardinality(n)][maxCardinality(m)] [cardinality(l)]															
Restriction T															
allValuesFrom(D) someValuesFrom(D)															
hasvalue (v)				•											
[minCardinality(n)] [maxCardinality(m)] [cardinality(l)]															
Data Ranges															
oneOf($v_1 \ldots v_n$)															

Table 1 OWL DL constructors used for class description in the tests

	Test														
# Test	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Success (S) or Failure (F)	S	S	S	S	S	S	S	S	S	S	S	F	F	F	F
Class Axioms															
EnumeratedClass(A $o_1 \ldots o_n$)															
DisjointClasses($C_1 \ldots C_n$)															
EquivalentClasses($C_1 \ldots C_n$)		•	•	•	•	•	•	•	•			•	•		
SubClassOf(C_1 C_2)		•	•	•	•	•	•	•	•	•	•	•	•	•	•
Property Axioms															
DatatypeProperty(T)															
[Functional]				•						•		•	•		
domain(C) range(D)				•						•		•	•		
EquivalentProperties($T_1..T_n$) SubPropertyOf(T_1 T_2)															
ObjectProperty(R)															
[inverseOf(R)]								•				•	•	•	
[Functional]									•		•	•	•		
[InverseFunctional]															
[Symmetric]												•	•		•
[Transitive]		•													
domain(C) range(D)		•	•	•	•	•	•	•	•	•	•	•	•	•	•
EquivalentProperties($R_1..R_n$)SubPropertyOf(R_1 R_2)															
Facts															
Individual(o)				•					•			•	•		
SameIndividual($o_1 \ldots o_n$)															
DifferentIndividuals($o_1 \ldots o_n$)				•					•			•	•		

Table 2 OWL DL Axioms and Facts used in the tests

3.3 Benefits

Although problems with computational resources occurred while reasoning with the full-fledged FMA in OWL DL, Racer could handle various less complex versions, which still enabled to detect *inconsistencies* in the original FMA and to *reclassify* some classes.

No inconsistencies were found in the first tested versions, but when datatype properties were added several inconsistencies were identified. 113 classes were identified as unsatisfiable by Racer because of opposite boolean values:

– *Inconsistencies from conflicts between metaclass and class definitions in Protégé.* A class having a boolean value in its own slot and which inherits the opposite value from its superclasses is unsatisfiable in OWL. For example, `Zone_of_cell` is unsatisfiable (hence, so are all its subclasses) because its own slot `has_mass` was assigned `false` at instance (converted to the restriction `has_mass:false`), while this single-slot had value `true` at its superclass `Material_physical_anatomical_entity` (converted to `has_mass:true`). Other inconsistencies were revealed from the inconsistency of the metaclass and class definitions of an entity. A class *A* subclass of *B* and instance of *C* in FMA, where *B* and *C* have opposite values for a boolean datatype property, e.g. `has_mass`, is unsatisfiable in OWL. For example, `Compartment_subdivision` is defined as a subclass of `Anatomical_cluster`, which is a subclass of `Material_physical_anatomical_entity` (`has_mass:true`). On the other hand, `Compartment_subdivision` is an instance of `Anatomical_space`, which is a subclass of `Non-material_physical_anatomical_entity` (`has_mass:false`).

– *Inconsistencies from conflicting global and local domain (or range).* `rdfs:range` (resp. `domain`) restrictions are global. Thus if p has class A' as domain and B' as range, and A has a property p with range B, then B must be a subclass of B' and A must be a subclass of A'. Conflicting definitions of global and local ranges or domains lead to inconsistencies in OWL. For example, `Surface_of_wrist` is unsatisfiable because '2D_part' has an existential restriction to `Anatomic_snuff_box` which is a subclass of `Material_physical_anatomical_entity` (`has_mass:true`), while the range of "2D_part" is `Non-material_physical_anatomical_entity` (`has_mass:false`). These inconsistencies exhibit modeling errors in the original Protégé FMA.

Racer also reclassified some classes. In the ontology including defined classes based on the `constitutional_part` property, 286 classes of the asserted hierarchy were moved within the inferred hierarchy and some classes were identified to be equivalent. For example, as the two sibling classes `Wall_of_biatrial_part_of_heart` and `Wall_of_biventricular_part_of_heart`,

have the same constitutional parts[6] in the original FMA, they became equivalent for this definition. However, the equivalence did not hold anymore when adding other restrictions to these definitions. For example, adding restrictions on the property `constitutional_part_of` enables to differentiate the two classes, as they are parts of different wholes: `Wall_of_biventricular_part_of_heart` is a `constitutional_part_of` `Biventricular_part_of_heart`, while `Wall_of_biatrial_part_of_heart` is a `constitutional_part_of` `Biatrial_part_of_heart`. Thus, although most of the reclassifications were related to the class definitions in terms of their constitutional parts, it nevertheless shows the power of reasoning with OWL DL.

In conclusion, the results obtained so far illustrate the benefits of representing the FMA in OWL DL. First, checking the logical consistency of the FMA enabled to find errors that would have probably been missed otherwise. Second, automatically computing the classification hierarchy is another advantage for such a large ontology. As the FMA has been under development at the University of Washington since 1994 and is still evolving, such services are very useful for quality assurance purposes.

4 Discussion and perspectives

This is a first step towards the representation of the FMA in OWL. Several issues remain open and different perspectives shall be considered in the future.

4.1 Using OWL DL or OWL Full ?

A recent study [5] proposes to translate the FMA into OWL Full, pointing out both computational and representational issues: "*The conversion from frames to OWL-DL required us to forgo representing some features of the FMA such as metaclasses …OWL-DL representation is possible, but requires to give up some of the original features*". The authors advocate selecting OWL Full for the FMA and OWL DL only for applications, suggesting a two-layered approach: *"The first layer consists of a generic conversion tool that generates a representation of the FMA in OWL-Full. The second layer consists of several application specific optimization tools that simplify the OWL Full representation of concepts into OWL-DL ones by removing all the features unnecessary according to the application context"*.

In contrast, we propose to use OWL DL for representing the FMA, and, more generally, reference ontologies. This work shows that converting the FMA from its frame-based representation in Protégé into OWL DL is indeed possible and succeeds in capturing most features of the original FMA: we converted the FMA into OWL DL with all the knowledge encoded at its metaclasses, and our conversion still complies with OWL DL constraints. In particular, no entity is at the same time both a class and

[6] Fibroelastic_connective_tissue_of_endocardium Fibrocollagenous_sheath_of_cardiac_muscle_tissue Fibroelastic_connective_tissue_of_epicardium

an individual. The original direct subclasses, superclasses, template slots, slot-constraints, defined at Protégé metaclasses are translated using OWL DL constructs and axioms. The main transformation that enabled to use OWL DL is the deletion of the Protégé higher order structure, achieved by replacing metaclass instantiations by subclass axioms. But this transformation did not introduce significant changes, since the class and the metaclasses hierarchies were integrated in the original model: "except for its root, all concepts in the Anatomy Taxonomy are subclass of a superclass and also an instance of a metaclass" [3]. As shown below, OWL DL's higher expressiveness enables to capture most intended meanings of the Protégé metaclasses (with the exception of "set of sets", but which does not represent a significant loss in our opinion, considering their use in Protégé). In fact, metaclasses were used for the following reasons [3] [4]:

(1) Metaclasses are used to model a given anatomical entitiy as a "set of sets", (e.g., `Vertebra` as a set of different types of vertebrae: cervical, thoracic, lumbar, themselves sets of other sets e.g., first, second, … , fifth lumbar vertebra). A first order language as OWL DL cannot capture this feature. However, the use of the representation of an entity as a "set of sets" is quite limited in Protégé. In fact, the "members of each of these collections are represented in Protégé as *subclasses* of Vertebra" [3] e.g., "the class `Vertebra` subsumes different collection of vertebrae, cervical, thoracic, and lumbar vertebra", which are further refined into more specialized subclasses.

(2) Metaclasses are also used "to enforce slot value restrictions" [4]. In frames, a slot inherited can only be refined to subclasses of its initial range. For example, when `Cervical_Vertebra` inherits from `Vertebra` the slot `part_of` with range `Vertebral_Column`, its range must be a subclass of `Vertebral_Column`. Metaclasses were intended to enforce restrictions to other classes, such as class `Cervical_Vertebral_Column`, which is not a subclass of `Vertebral_Column` in the FMA model, but a part of it. Thanks to metaclass instantiation, the appropriate values are assigned to own slots at class (§ 2.1). This artefact is no longer needed in OWL, since it is possible to use `subClassOf` axioms with various restrictions instead, e.g., ∃ `part_of Vertebral_Column` for the class `Vertebra` and ∃ `part_of Cervical_Vertebral_Column` for its subclass `Cervical_Vertebra`, although `Cervical_Vertebral_Column` is not subsumed by `Vertebral_Column`.

(3) Metaclasses were also intended to specify multiple values specific to each class e.g., specifying that a `Vertebra` has parts `Body_of_vertebra`, `Vertebral_arch`, `Bone_of_vertebra`, etc. In OWL, this can be captured by the conjunction of several restrictions such as (∃ `part_of Body_of_vertebra`) ⊓ (∃ `part_of Vertebral_arch`) ⊓ (∃ `part_of Bone_of_vertebra`) etc.

(4) Finally, metaclasses are used for specifying metadata such as name, author, authority, UWDAID, etc. Assigning values to these 'non structural' own slots at metaclass instantiation prevents them from being propagated to their instances or subclasses. This can be obtained in OWL in using `AnnotationProperty`.

The main reason for selecting OWL DL over OWL Full is that OWL DL is decidable and supports powerful reasoning services such as consistency checking and automatic classification. OWL DL reasoners are available (e.g., Racer [16], Pellet [17]), while OWL Full is undecidable, offers no computational guarantees and lacks suitable reasoners. A reference ontology is generally a large scale ontology intended to provide a *controlled* vocabulary for a domain. Thus, it is crucial to guarantee the quality and correctness of such reference ontologies sharable on the Web. Automated reasoning services are precious for it. Current results are encouraging and the computational difficulties encountered during this study will undoubtedly be overcome soon. In any case, as shown in §3.2 - 3.3, the results inferred from reasoning even on partial versions, are helpful for improving the consistency of the entire ontology.

4.2 Scalability of reasoning

The size and complexity of the FMA raise important computational issues. As far as we know, the NCI Thesaurus was one of the largest file in Protégé OWL so far. But it is much smaller and exhibits less complexity than the FMA in OWL. The NCI Thesaurus contains 53,000 frames, including 34,000 classes, 100 properties and 9,000 conditions, while the original FMA contains 70,000 concepts and the converted subset 117,000 frames, including 40,000 OWL classes, 187 properties and about 110,000 axioms. But the most important difference is the language complexity. The NCI Thesaurus was converted to OWL Lite, while the FMA is represented in OWL DL. No defined class, no `hasValue`, `allValuesFrom restrictions`, `nor unionOf or nominals` (enumerated classes `oneOf`) are specified in the NCI Thesaurus, while they all occur in the FMA in OWL DL. The size and complexity of the FMA in OWL make it a real challenge for DLs systems. The current tests done with the FMA show that, with the current state of the art of DL inference technology, such complexity might result in inference problems hard to solve in terms of time and space resources. Indeed, the main problem was computational. Some optimizations were devised to reduce the complexity of the generated file. For example, it was necessary to reduce the number of disjunctions generated by the conversion for the domain of properties, which otherwise caused Racer – or any inference system – to run into space problems. Interestingly, after optimization, two classes remain in the domain of `location` instead of 1,618 originally [15]. Difficulties also occurred for inverse properties. However, Racer could handle various less complex versions of the FMA in OWL DL, detect inconsistencies, and reclassify classes. This experiment was done with Racer version 1.7. As Racer evolves – for example, its authors are currently working on optimizations that address the issue of inverse roles – it will be worthwhile to repeat these tests with the next versions and also to evaluate the performance of other OWL DL reasoners (e.g., Pellet [17] Fact++ [18]). The results obtained with RacerPro™ [16] are encouraging so far. The new decision procedures and optimizations being currently implemented in highly optimised DL reasoners, e.g., Fact++ [18] may also improve the results in a next future.

4.3 Domain and application ontologies

A reference ontology such as the FMA, is a 'domain' ontology supposed to be application-independent. Applications most often do not require the whole FMA but only a specific module extracted from it. What is more problematic is that different applications may require different class definitions, because of their specific goals. Indeed, results of reasoning with OWL DL strongly rely on the defined classes expressions, and the intended use of the ontology biases the N&S conditions from what is to be achieved via classification or instance recognition. For example, the Brain Cortex Anatomy application [7] only needs concepts involved in Brain anatomy and instance recognition of some entities is mainly based on their definitions in terms of boundaries. Another application, like the Virtual Soldier project may require concepts involved in the neighborhood of the heart and perhaps mainly definitions based on parts. Hence, the question that arises is whether it is achievable to have 'application-independent' class definitions for a reference ontology such as the FMA.

In fact, several possible options might be considered for the defined classes of the FMA (1) each class has a single definition, based on the conjunction of all the qualified property restrictions derived from the values of its own structural slots and attributed relations; (2) each class has a set of several equivalent definitions; (3) each class has one preferred definition, the other conditions being simply necessary; (4) there are no a priori "defined" classes but only primitive OWL DL classes, all axioms expressing only necessary conditions. As the FMA aims at being a "shared reference ontology", its representation in OWL DL might be considered as *a first formal specification*, to be further refined into more *detailed formal specifications* for each application, in adding relevant axioms. Therefore, the best options to consider for large-scale domain ontologies such as the FMA, might be to represent them in OWL DL either with a preferred definition (3) or with only primitive classes (4), but with a library of optional usual class equivalent definitions, the validity of which having been checked. Then, more specific OWL DL ontologies may be further customized for particular applications, e.g., by adding relevant equivalent class axioms (selected from the library or created specifically). The advantage of this solution is twofold. First, it would concretely implements the notion of a "Semantic Web reference ontology" specified independently of applications. Second, it allows users to benefit from DL reasoning services such as consistency checking and classification for both the general reference ontology and the more customized ones.

4.4 Future work

The next step is to improve the current conversion so as to provide a more reliable and enriched representation of the FMA in OWL. A possible direction is to introduce additional enrichment to the FMA original model mainly by adding qualified number restrictions, disjointness and closure axioms (§4.4.1) and by creating reliable formal *definitions* for the classes (§4.4.2).

4.4.1 **Adding qualified number restrictions, disjointness and closure axioms**

In the future, we would like to improve the current conversion process and to remove some of its limitations:

- First, we suggest adding *disjointness axioms* between the relevant classes. Ideally, a classification satisfies the so-called "jointly exhaustive and pairwise disjoint" rule. The inconsistencies reported §3 are mainly based on opposite values of a boolean datatype property and their propagation, but disjointness axioms will most probably lead to identifying more inconsistencies in the FMA.
- Second, we propose using *qualified cardinality restrictions*. We converted structural own slots values by existential property restrictions, mainly for two reasons. On the one hand, the assumption that if a class *A* has a slot *p* filled with values $B_1, B_2 \ldots B_n$ in Protégé (e.g., constitutional part), an implicit assumption is that for every individual of *A*, *p* has at least one value of each class B_i. On the other hand we were confronted to the expressiveness limitation of OWL, which does not support qualified cardinality restrictions. However, defining restrictions such as "has Part `someValuesFrom` B_1" and "hasPart `someValueFrom` B_2" is weaker than "hasPart exactly one B_1 and one B_2", as it does not prevent from having several parts of the same B_i. If the next version of OWL supports qualified cardinality restrictions, more precise definitions might be provided.
- Thirdly, we suggest completing our current class definitions by so-called *closure axioms* [13]. In fact, neither existential property restrictions nor qualified cardinality restrictions, prevent properties from being assigned values from an unwanted class. In contrast, adding a `allValuesFrom` restriction to the class $B_1 \sqcup B_2$ would coerce values to come *only* from B_1 or B_2.

  Qualified cardinality and closure axioms would allow to reflect more closely the original FMA definitions, either given in natural language or implicit in Protégé frames. For example, the equivalent class definition `Left_lung` ≡ `Lung` ⊓ (= 1 `regional_part Upper_lobe_of_left_lung`) ⊓ (= 1 `regional_part Lower_lobe_of_left_lung`) ⊓ (∀ `regional_part Upper_lobe_of_left_lung` ⊔ `Lower_lobe_of_left_lung`) ⊓ etc. would enable the Left Lung to be defined as having exactly one left upper lobe, one left lower lobe and only those two lobes as regional parts, or a Right Lung as having exactly one right upper lobe, one middle lobe and one right lower lobe and only those three lobes, reflecting the original definitions from the Protégé FMA:

```
([Left_lung] of Lung
    (definition "Lung which consists of the left upper
      lobe and left lower lobe" )
    (regional_part
           Upper_lobe_of_left_lung
           Lower_lobe_of_left_lung)
    …)
([Right _lung] of Lung
    (definition "Lung which consists of the right upper
      lobe, middle lobe and right lower lobe")
    (regional_part
           Upper_lobe_of_right_lung
           Middle_lobe_of_lung
           Lower_lobe_of_right_lung)
    …)
```

4.4.2 **Enriching the FMA with formal definitions**

Currently, the biggest challenge in converting the FMA from its original Protégé frame-based representation to OWL DL is certainly to specify explicit and precise formal definitions for the classes, i.e., necessary and sufficient conditions. In the current experimental version, the equivalent class expressions used to define classes combine taxonomic relations, metaclass instantiation and constitutional parts that were defined in the original FMA model. Only one property, `constitutional_part` (or `custom_partonomy`) was selected for the equivalent class definitions. This simple method is not satisfying in many respects and is a serious limitation for the results obtained from reasoning, especially from classification. Defining anatomical entities solely on the basis of their constitutional parts is not semantically correct for all classes, and no such constitutional parts are defined for most FMA classes. Defining classes from their constitutional parts may perhaps be appropriate for Organ, but other anatomical entities like Organ part, Cell, or Tissue etc. need different criteria of identification. As all the anatomical entities do not share the same kind of definition, different expression templates should be specified for the different subtrees, e.g., `Organ, Cell`, etc. Equivalent conditions – single or multiple, default or optional – must be defined in close collaboration with the FMA authors, so as to have "semantically" correct expressions supporting the unique identification of anatomical entities. Conversion rules should be modified to accommodate arbitrary combinations of properties, constructors, and cardinality restrictions and to support specific expressions for each subtree.

At this first stage, the conversion aimed at capturing the features of the Protégé representation of the FMA as faithfully as possible, in order to evaluate its original properties. In the future, in addition to above proposals, we suggest to introduce some changes in the model. For example, the OWL classes used for the Protégé attributed relations might be specified by *n*-ary relations in an external base related to the ontology. New classes might be defined for structuration and consistency reasons, e.g., `Venous_drainage`, `Arterial_Supply`, etc.

5 Conclusion

Converting the whole FMA from its original frame-based representation into the first order language OWL DL was possible, and most features of the original FMA model were captured. With the current state-of-art of DL reasoners, reasoning with OWL proved to be a real challenge, because of the sheer size and complexity of the FMA in OWL. Reasoning with the full-fledged FMA raised computational difficulties, but after some optimizations, various smaller versions were successfully tested with Racer. Several inconsistencies were revealed in the original modeling of the FMA. Some classes of the asserted hierarchy were reclassified and some classes identified to be equivalent. Although most of them were related to the definition of the anatomical

entities in terms of their constitutional parts, it illustrates the power of reasoning with OWL DL. The results obtained so far demonstrate the advantages of OWL over frames for large-scale domain ontologies of the future Semantic Web such as the FMA. However, this experiment is only a first step and the current versions are not yet satisfactory in many respects. In particular, the experimental method used for creating defined classes from their constitutional parts has to be revised. Correct expressions supporting the unique identification of anatomical entities shall be defined. The conversion rules shall be modified and refined. Tests with RacerPro™ are encouraging. Other reasoners also need to be tested. OWL semantics and DL techniques have the potential of providing a significant help in enriching and improving the FMA ontology for the future Semantic Web of Life Sciences.

Acknowledgments

This research was supported in part by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM), while Pr. Christine Golbreich and Dr. Songmao Zhang were visiting scholars at the Lister Hill National Center for Biomedical Communications, NLM, NIH. Our thanks to Volker Haarslev and Ralf Möller for their valuable advice and help on OWL and Racer, to Cornelius Rosse, José Mejino and Todd Detwiler for the FMA, and to other helpful advice.

References

1. Mike Dean, Guus Schreiber, Sean Bechhofer, Frank van Harmelen, Jim Hendler,Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL web ontology language reference. W3C Recommendation, 2004. Available at http://www.w3.org/TR/owl-ref/
2. Cornet R, Abu-Hanna A. Usability of expressive description logics--a case study in UMLS. ProcAMIA Symp 2002:180-4
3. Rosse C, Mejino JL, Jr. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. J Biomedical Informatics 2003; 36(6):478-500.
4. Noy N, Musen MA, Mejino JL and Rosse C. Pushing the Envelope: Challenges in a Frame-Based Representation of Human Anatomy. Data and Knowledge Engineering Journal; 2002 48(3):335-359.
5. Dameron O, Rubin DL and Musen AM. Challenges in Converting Frame-Based Ontology into OWL: the Foundational Model of Anatomy Case-Study. Proc. AMIA Annual Symposium 2005:181-185
6. Golbreich C., Zhang S., Bodenreider O., Migrating the FMA from Protégé to OWL. 8th Protégé Intern. Conf., Madrid 2005:108-111
7. Golbreich C., Dameron O. , Bierlaire O., Gibaud B. What reasoning support for Ontology and Rules? the brain anatomy case study. Proceedings of the workshop "OWL Experiences and Directions", Nov 11-12, 2005, Galway, Ireland 2005: electronic proceedings: http://CEUR-WS.org
8. Soualmia L, Golbreich C, Darmoni S. Representing the MeSH in OWL: Towards a semi-automatic migration. Proceedings of the KR 2004 Workshop on Formal Biomedical Knowledge Representation 2004:81-87, available at http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-102/soualmia.pdf

9. Wroe CJ, Stevens R, Goble CA, Ashburner M. A methodology to migrate the Gene Ontology to a description logic environment using DAML OIL. Pacific Symposium Biocomputing 2003:624-35
10. Golbeck J, Fragoso G, Hartel F, Hendler J, Oberthaler J, Parsia B. The National Cancer Institute's thesaurus and ontology. Journal of Web Semantics 2003;1(1)
11. Kashyap V, Borgida A. Representing the UMLS Semantic Network using OWL: (Or "What's in a Semantic Web link?"). In: Fensel D, Sycara K, Mylopoulos J, editors. The SemanticWeb – ISWC 2003. Heidelberg: Springer-Verlag; 2003. p. 1-16
12. Noy N. F. Sintek M, Decker S., Crubezy M, Fergerson R. W., & Musen M. A.. Creating Semantic Web Contents with Protege-2000. IEEE Intelligent Systems 16(2): 60-71, 2001.
13. Rector A, Drummond N, Horridge M, Rogers J, Knublauch H, Stevens R, Wang H, Wroe C. OWL Pizzas: Practical Experience in Teaching OWL-DL: Common Errors and Common Patterns. European Conference on Knowledge Acquisition. 63-81. 2004.
14. Rosse C., Kumar A. , Mejino J.L., Cook D., Detwiler L., Smith B., A Strategy for Improving and Integrating Biomedical Ontologies, Proceedings of AMIA Symposium 2005, Washington, 2005: 639-643.
15. Zhang S, Bodenreider O, Golbreich C. Experience in reasoning with the Foundational Model of Anatomy in OWL DL. Proc. Pacific Symposium on Biocomputing 2006 (in press).
16. RacerPro: http://www.franz.com/products/racer/
17. Pellet OWL reasoner. Maryland Information and Network Dynamics Lab, 2003. http://www. mindswap.org/2003/pellet/index.shtml.
18. Tsarkov D. and Horrocks I. Efficient reasoning with range and domain constraints. In Proc. of DL 2004, p. 41–50, 2004.

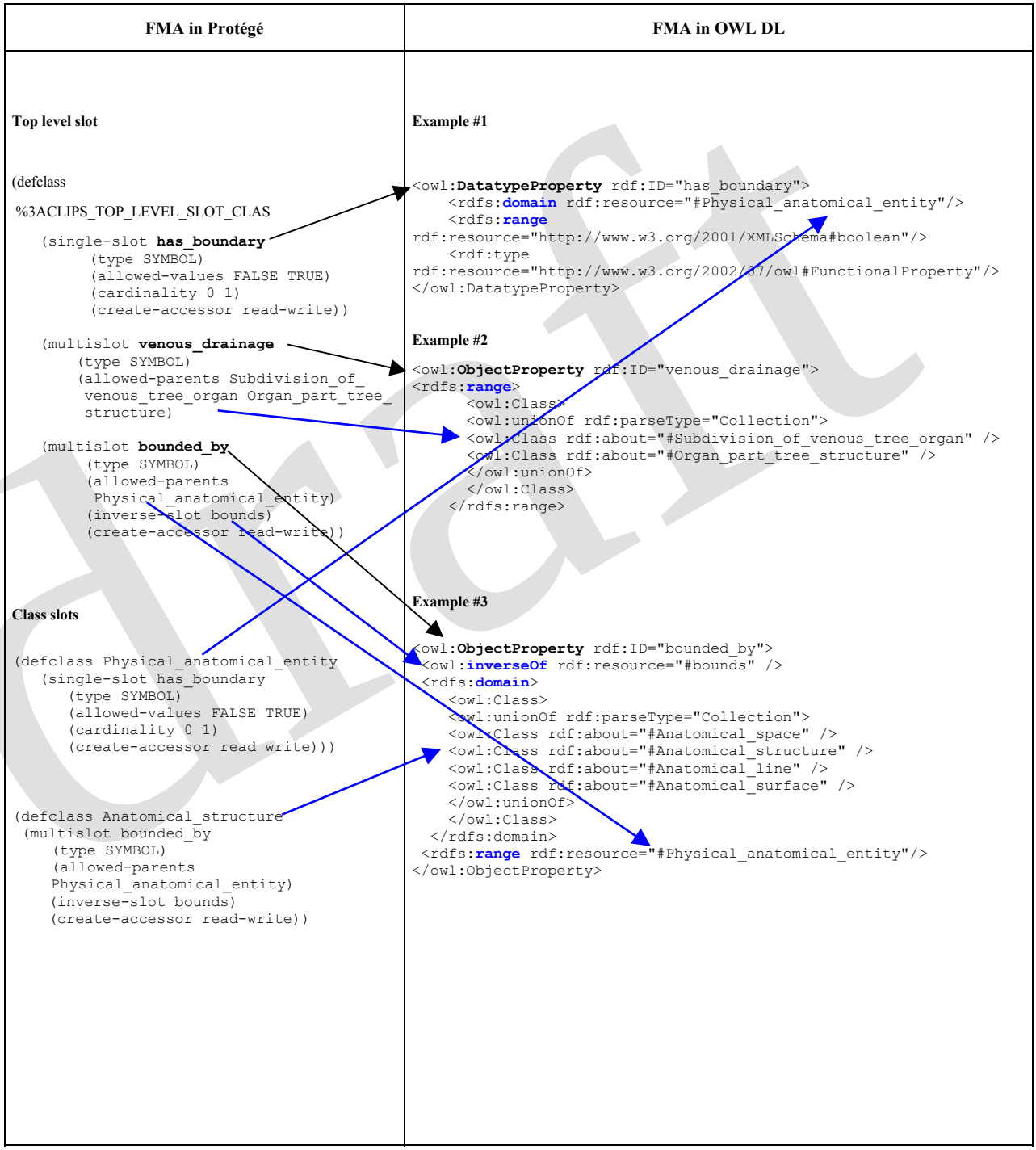
ANNEX

Conversion rules

The rules depend on whether the information is stored at metaclass or class level. They capture all the knowledge of the original Protégé model, defined either at metaclasses, classes, or instances, while respecting the original principles of the FMA.

1. Class information. Classes and slots – stored at (meta)class level in Protégé – are converted into OWL classes and properties with specified domain (`rdfs:domain`) and range (`rdfs:range`). The property characteristics are translated to OWL corresponding constructs: inverse to `owl:inverseOf`, logical characteristics, i.e. symmetric to `owl:SymmetricProperty`, property cardinality and values restrictions to `owl:FunctionalProperty`, and `owl:hasValue`. In practice, the rules are the following:

- **Top level slots**, specified in Protégé to save top-level slot information, are converted to `DatatypeProperty` or `ObjectProperty` with the corresponding range and restrictions, according to their definition. For example,
  (1) a top-level slot with type SYMBOL and allowed-values TRUE FALSE is converted to a `DatatypeProperty` with `range #Boolean` e.g., `has_boundary` (**Table 3** Example #1)
  (2) a top-level slot with type SYMBOL with allowed-values different from TRUE FALSE is converted to an `ObjectProperty` with an enumerated class `oneOf`{allowed-values} as range
  (3) a top-level slot with type SYMBOL with allowed-classes (allowed-parents) is converted to an `ObjectProperty` with the union of the allowed (meta)classes as `range`, e.g., the `range` of the multislot `venous_drainage` is the `unionOf #Subdivision_of_venous_tree_organ` and `#Organ_part_tree_structure` (Example #2)
  (4)a top-level slot with type INSTANCE is converted to an `ObjectProperty`
  Single-slot with cardinality 0 1 is converted to `FunctionalProperty` (Example #1).
  Inverse-slot. If a top level slot has an 'inverse-slot', it is converted to `SymmetricProperty` or `InverseOf`: if the inverse value is itself, it is `SymmetricProperty` with range assigned to its domain, else it is `InverseOf`.
  Thus, for example, the top level slot `has_boundary` is converted to a `DatatypeProperty` with `range #boolean`, with a `FunctionalProperty` restriction, while `bounded_by` is converted to an `ObjectProperty` with `range #Physical_anatomical_entity`, and `inverseOf #bounds` (Example #3).


FMA in Protégé
FMA in OWL DL
Top level slot
(defclass
%3ACLIPS_TOP_LEVEL_SLOT_CLAS
(single-slot has_boundary
(type SYMBOL)
(allowed-values FALSE TRUE)
(cardinality 0 1)
(create-accessor read-write))
(multislot venous_drainage
(type SYMBOL)
(allowed-parents Subdivision_of_
venous_tree_organ Organ_part_tree_
structure)
(multislot bounded_by
(type SYMBOL)
(allowed-parents
Physical_anatomical_entity)
(inverse-slot bounds)
(create-accessor read-write))
Class slots
(defclass Physical_anatomical_entity
(single-slot has_boundary
(type SYMBOL)
(allowed-values FALSE TRUE)
(cardinality 0 1)
(create-accessor read write)))
(defclass Anatomical_structure
(multislot bounded_by
(type SYMBOL)
(allowed-parents
Physical_anatomical_entity)
(inverse-slot bounds)
(create-accessor read-write))
Example #1
<owl:DatatypeProperty rdf:ID="has_boundary">
<rdfs:domain rdf:resource="#Physical_anatomical_entity"/>
<rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
<rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
Example #2
<owl:ObjectProperty rdf:ID="venous_drainage">
<rdfs:range>
<owl:Class>
<owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#Subdivision_of_venous_tree_organ" />
<owl:Class rdf:about="#Organ_part_tree_structure" />
</owl:unionOf>
</owl:Class>
</rdfs:range>
Example #3
<owl:ObjectProperty rdf:ID="bounded_by">
<owl:inverseOf rdf:resource="#bounds" />
<rdfs:domain>
<owl:Class>
<owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#Anatomical_space" />
<owl:Class rdf:about="#Anatomical_structure" />
<owl:Class rdf:about="#Anatomical_line" />
<owl:Class rdf:about="#Anatomical_surface" />
</owl:unionOf>
</owl:Class>
</rdfs:domain>
<rdfs:range rdf:resource="#Physical_anatomical_entity"/>
</owl:ObjectProperty>

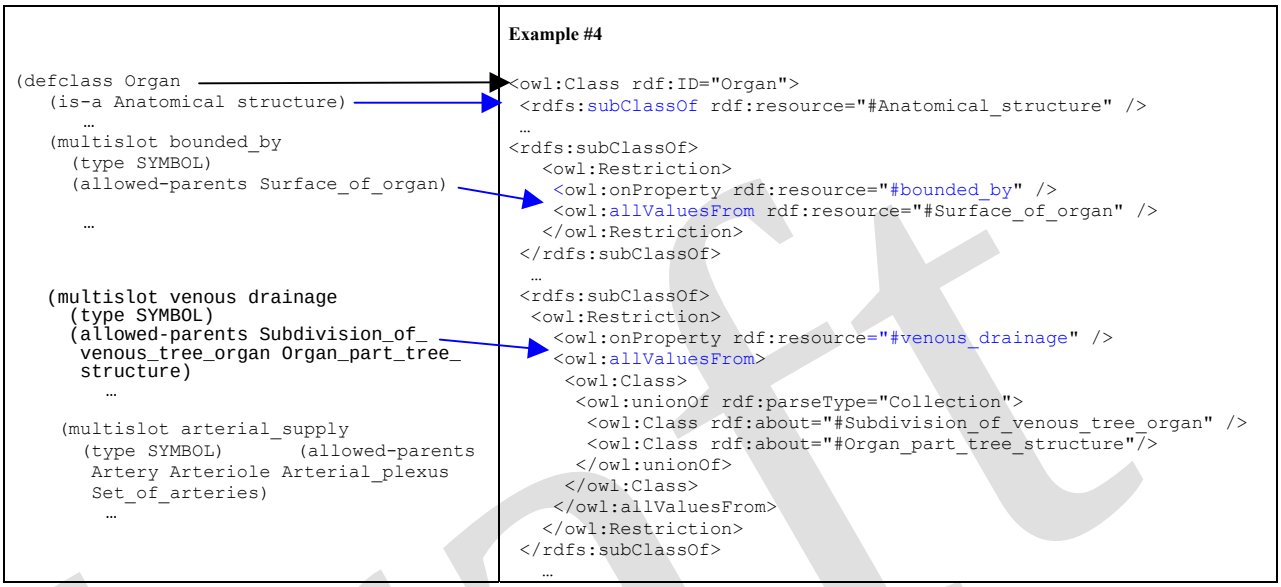


Table 3 Examples of conversion rules for top level and template slots

– **Slots at class** enable to define the `domain` of an OWL property and to refine its value constraints: if p is slot of class C1, then C1 becomes the domain of p e.g. `#Physical_anatomical_entity` becomes the domain of `has_boundary` (Example #1); if the same slot p occurs in class C2, then the domain of p is the union of C1 and C2 (e.g. the domain of `bounded_by` in Example #3); Optimization of domain C1 ⊔ C2… ⊔ Cn has been done: if Ci is descendant of another class according to two levels of is-a, then Ci is removed from the domain (reducing so "arterial supply" domain from 4007 classes to 4!).
Allowed-parents, allowed–classes, value define the allowed values of properties at class. They are converted to necessary conditions expressing value constraints on the property for this class: if p is slot of class C specified with allowed-parents or allowed-classes (resp. with value), then p is converted by a necessary condition at class C expressing value constraints for p by `owl:allValuesFrom` the union class of all its allowed-parents or allowed-classes e.g., `allowed-parents Surface_of_organ` or `allowed-parents Subdivision_of_venous_tree_organ Organ_part_tree_structure` (Example #4) (resp. by `hasValue`).
Is-a is converted into subsumption axioms (`subClassOf`): A is-a B (if B is not USER nor :STANDARD-CLASS or :STANDARD-SLOT or RELATION) is converted to A `subClassOf` B (resp. A is-a B1 and A is-a B2 is converted to `subClassOf` B1 ⊓ `subClassOf` B2), e.g. `Organ is-a Anatomical_structure` (Example #4).

2. Instance information. The values of class own slots – specified at instance level in Protégé to store data specific to a class – are converted either into OWL values of annotation properties or into existential restrictions on the class properties. In practice, the rules are the following:

– **Non structural slots**. In Protégé non structural slots such as 'preferred name', 'synonyms', 'definition', 'UWDAID', 'author' etc., are defined as slots of metaclasses. When classes instantiate the metaclasses, they become own slots assigned with values specific to each class, which are thus not propagated to their instances or subclasses [4]. For example, UWDAID value for Heart is 7088. We used OWL annotations on classes instead, which are allowed in OWL-DL under some constraints e.g., the AnnotationProperty UWDAID (Table 4). Properties designed as AnnotationProperty have been manually selected. They include Preferred_name, Synonyms, UWDAID, author, authority, modification, name, Date_entered_modified, TA_ID, definition, modification, "Latin_name_TA", UMLS_ID, Outdated_meaning, ther_Latin_equivalents, Source, View, Abbreviation English_equivalent, etc.

```
<owl:AnnotationProperty rdf:ID="UWDAID" />

<owl:Class rdf:ID="Heart">
   …
  <UWDAID>7088</UWDAID>
</owl:Class>
```

Table 4 Conversion of a non structural own slot into an AnnotationProperty

– **Structural slots**. Another use of metaclass in Protégé is for "structural" slots, such as part_of, custom_partonomy, bounded_by, arterial_supply, etc. It enables to specify each class for "canonical" anatomy with particular values assigned to its own slots, which are thus not propagated.

Heart in Protégé	Heart in OWL DL
([Heart] of Organ_with_cavitated_ organ_parts (bounded_by Surface_of_heart)	<owl:Class rdf:ID="Heart"> <rdfs:subClassOf rdf:resource="#Organ_with_cavitated _organ_parts"/> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="#bounded_by" /> <owl:someValuesFrom rdf:resource="#Surface_of_heart"/> </owl:Restriction> </rdfs:subClassOf>

Table 5 Conversion of a structural own slot into an existential property restriction

For example, a 'canonical' Heart is specified having a Right_atrium, a Left_atrium, a Right_ventricle, a Left_ventricule as

custom_partonomy, being bounded_by a Surface_of_heart. Structural own slots are converted to necessary (or necessary and sufficient) conditions at class C asserting someValuesFrom constraints for p to the union of all the classes assigned to p. For example bounded_by Surface_of_heart is converted to a someValuesFrom restriction on the property #bounded_by, expressing that any instance of heart is necessarily bounded by at least one #Surface_of_heart (Table 5).

- **Attributed relations**. The values of attributed relations are represented in OWL by nested classes generated in following the same conversion rules as for classes.

Heart in Protégé	Heart in OWL
([Heart] of Organ_with_ cavitated_organ_parts (attributed_part [fm-live_12491] [fm-live_12492] [fm_live_17313] [fm_live_17314] [fm_live_17315] [fm_live_17316] [fm_live_17317] [fm_live_17318] [fm_live_17319] [fm_live_17320] [fm_live_17321] [fm_live_17322] [fm_live_17323]) where ([fm-live_12491] of Organ _subdivision_part_of_ relationship_value (anatomical_arbitrary Arbitrary) (partition Partition_2) (related part Right_side _of_heart) (shared_unshared Unshared))	<rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="#attributed_part" /> <owl:someValuesFrom> <owl:Class rdf:ID="fm-live_12491"> <!-- nested class from [fm-live_12491] --> <rdfs:subClassOfrdf:resource="#Organ_subdivision _part_of_relationship_value" /> <rdfs:subClassOf> <owl:Restriction> <owl:onPropertyrdf:resource="#anatomical_arbitrary"/> <owl:hasValue rdf:resource="#individual_Arbitrary" /> </owl:Restriction> </rdfs:subClassOf> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="#partition" /> <owl:hasValue rdf:resource="#individual_Partition_2"/> </owl:Restriction> </rdfs:subClassOf> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="#related_part" /> <owl:someValuesFrom rdf:resource="#Right_side_of_heart"/> </owl:Restriction> </rdfs:subClassOf> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="#shared_unshared" /> <owl:hasValue rdf:resource="#individual_Unshared" /> </owl:Restriction> </rdfs:subClassOf> </owl:Class> <!-- end of nested class for instance [fm-live_12491] --> </owl:someValuesFrom> </owl:Restriction> </rdfs:subClassOf> <rdfs:subClassOf> … </owl:Class>

Table 6 Conversion of attributed relations

For example attributed_part is an attributed relation which allowed values are instances of class Part_of_relationship_value e.g. fm-live_12491, fm-live_12492 etc. They are converted to someValuesFrom restrictions on the property #attributed_part (Table 6).

– **Instantiation of metaclasses.** Metaclass instantiation is replaced by subsumption axiom using `subClassOf` axioms[7]: [A] of B is converted to a `subClassOf` axiom A ⊏ B e.g., the axiom of class `Heart subClassOf Organ_with_cavitated_organ_parts` Table 5.

– **Generating individuals in OWL DL.** Based on the slot `dimension` presented earlier, one individual[8] is generated under `owl:Thing` for each allowed value of this slot, as shown in Table 7

.

```
<owl:Thing rdf:ID="individual_0-dimension" />
<owl:Thing rdf:ID="individual_1-dimension" />
<owl:Thing rdf:ID="individual_2-dimension" />
<owl:Thing rdf:ID="individual_3-dimension" />
```

Table 7 Generating individuals in OWL DL

Additional examples illustrating the conversion rules are available online at http://mor.nlm.nih.gov/pubs/supp/2006-psbsz/2006-psb-sz-supp.pdf

[7] For "[A] of A", "[A] of B " when "A is-a B ", or "[A] of B " when "A is-a C" and C is a descendant of B, some optimizations prevent the generation of useless axioms.

[8] Individuals are prefixed by "individual_", because some allowed-values of slots share names with classes in the FMA in Protégé, such as *Inferior* and *Liquid*.